Intel 8xx series and Paxville Xeon-MP Microprocessors

Jonathan Douglas

Intel Corporation

Thanks to: Justin Marquart, James Vogeltanz, Mike Grassi, DEG/BCG package design, Donald Parker & Benson Inkley for help in putting together this presentation.

Outline

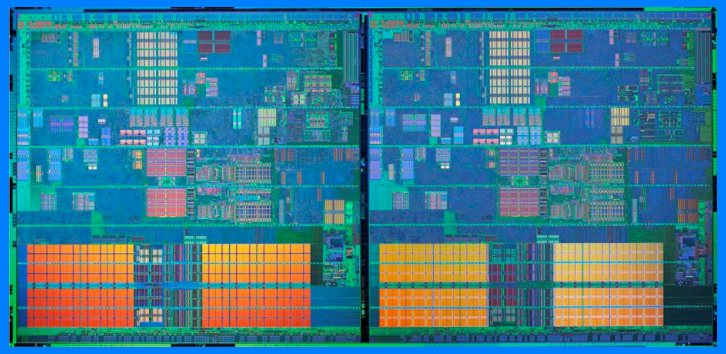
- Why the move to multi-core
- Overview of 8xx series Pentium4
- Challenges in moving CPU infrastructure to multi-core
- Learning's from the 8xx series Pentium4 design
- Overview of Paxville-MP processor
- Going forward with multi-core designs
- Conclusion

Why rapid move to Dual-Core

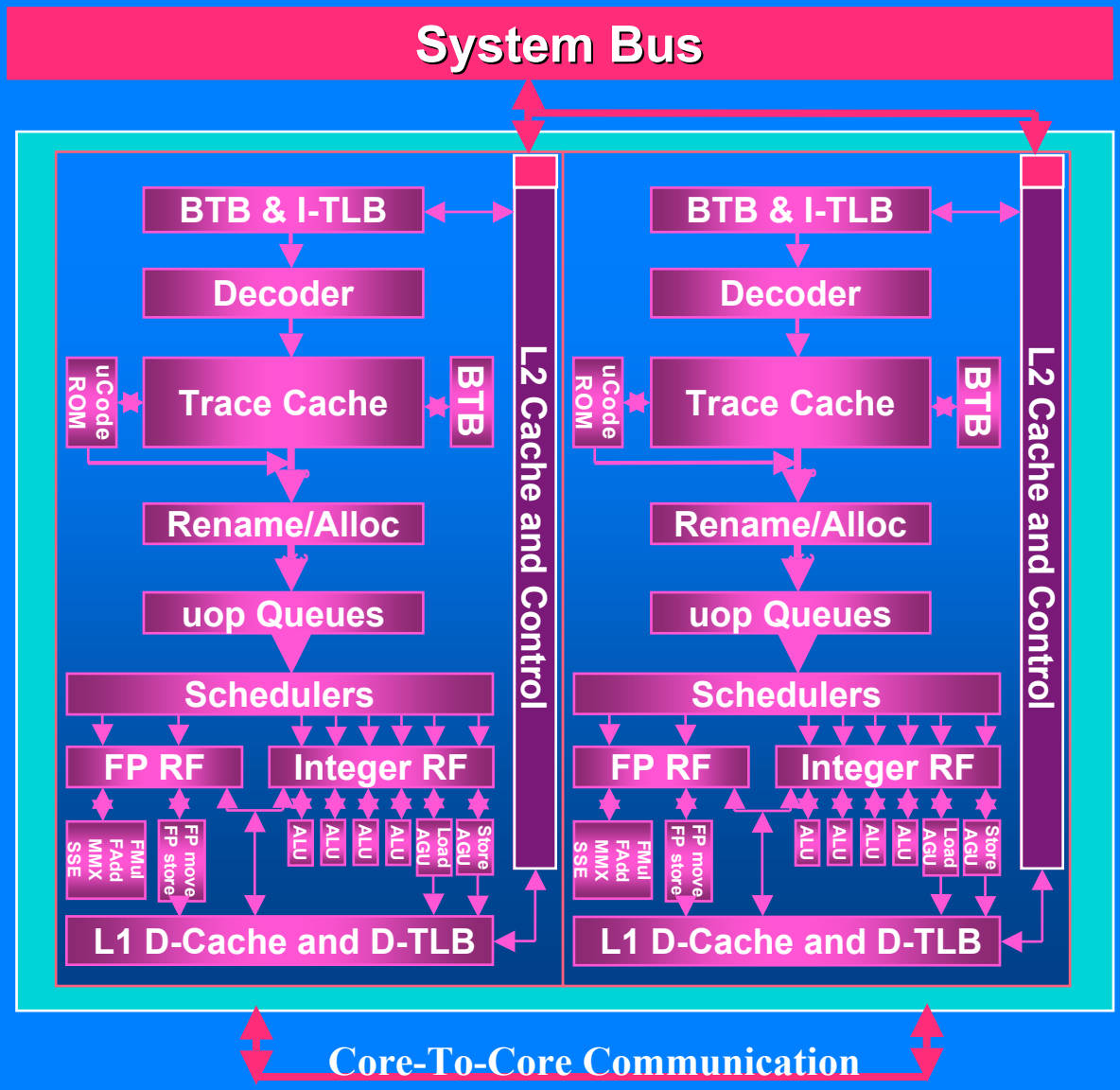
- Single core designs hitting power wall.
  - Need more power efficient way to manage OS loading.
- Natural extension of software migration to multi-threaded apps.
- More threads in 1 core is complex and tax core resources heavily.
- Competitive response.

Overview of 8xx series Pentium4 processor

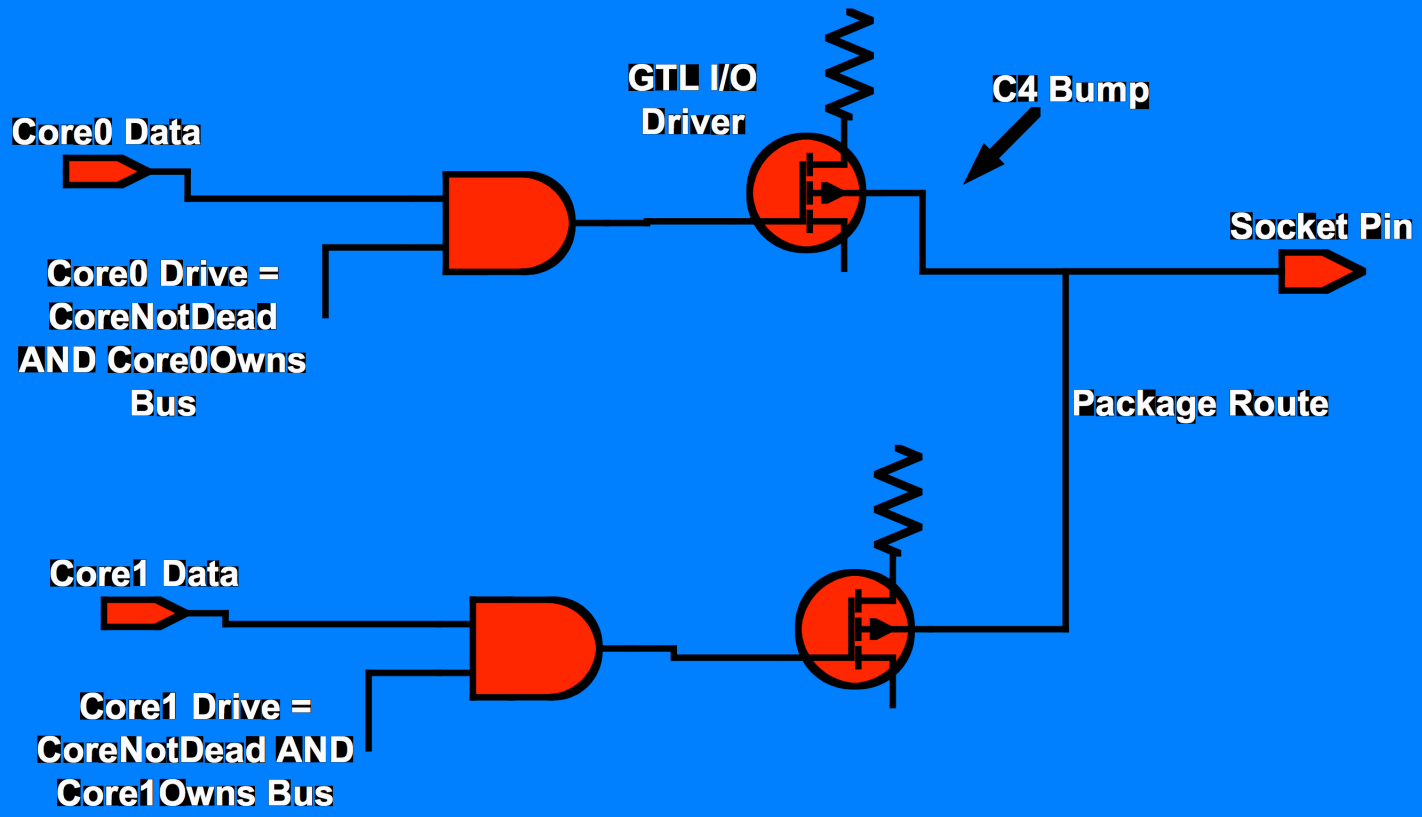
- Dual-Core/Multi-Threaded Pentium®4 Processor on 90nm process
  - 2-1M caches, speeds to 3.2Ghz, support for over clocking, up to 4 threads.
- Shared 800Mhz quad-pumped FSB.
  - Independent bus tuning per agent
- Enhanced auto-halt and 2-state speed step power management
  - Independent events supported per core.



High level block diagram



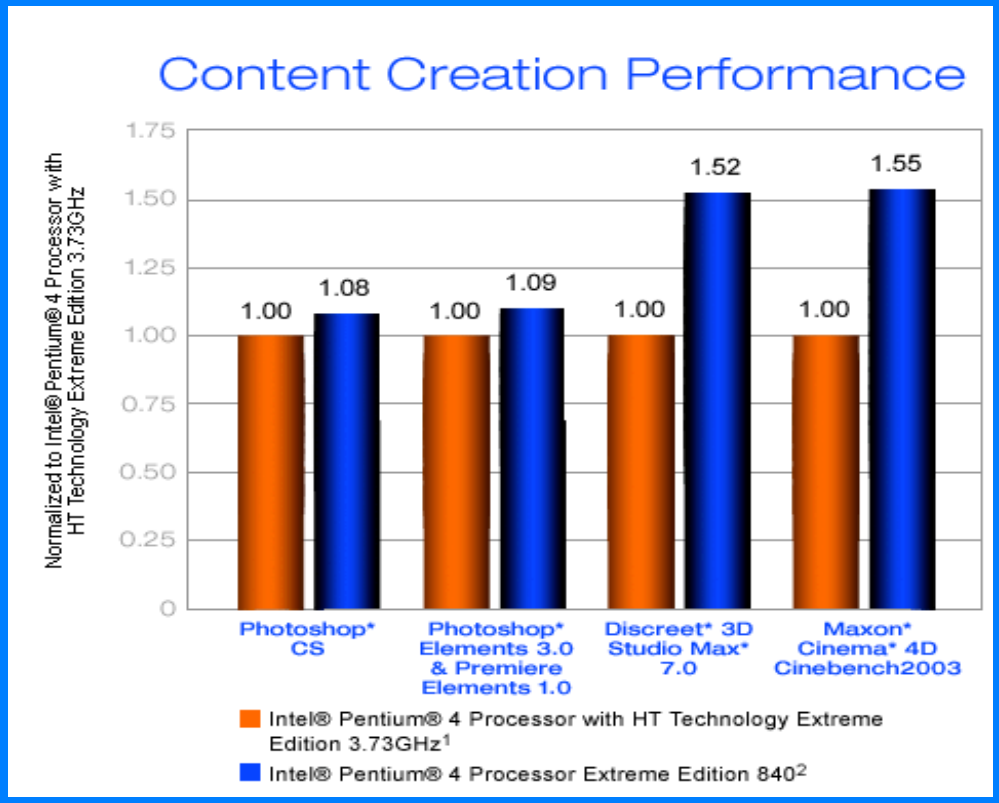
Zoom in on Front-Side Bus

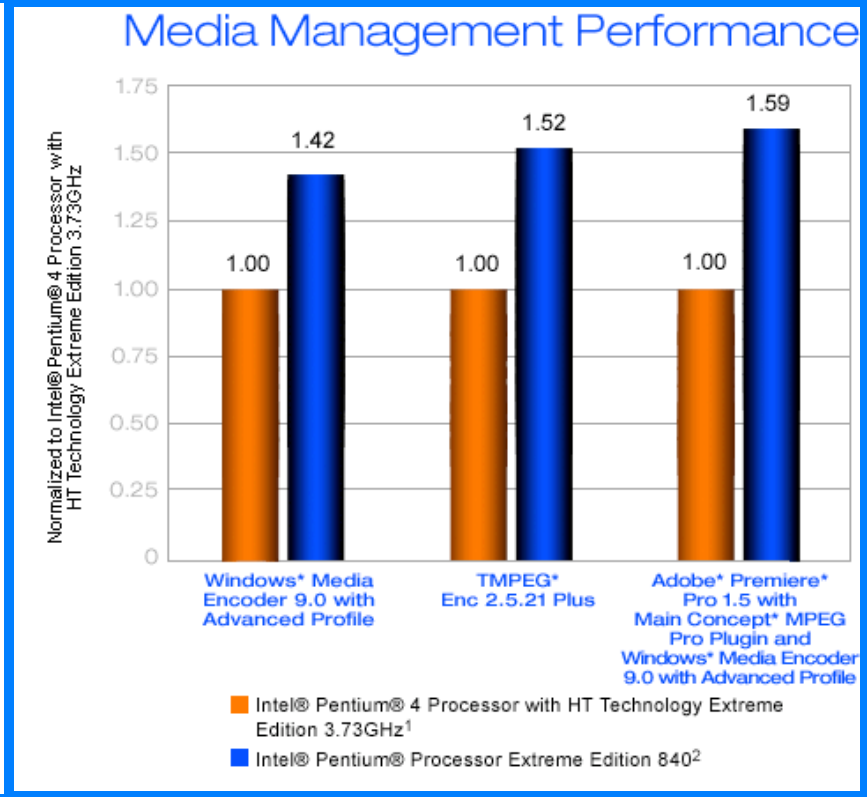


Why the shared bus design

- Time to market a critical factor
  - Leverages existing P4 core
  - Uses existing 775-LGA socket
- P4 core already has right feature set
  - P4 FSB already 4-way compliant.
  - Already architected with thread independent power management.
  - Already ‘HT’ so 2 cores = 4 threads
- Gives independent caches
  - Plus no extra latency to external memory.

Dual core performance





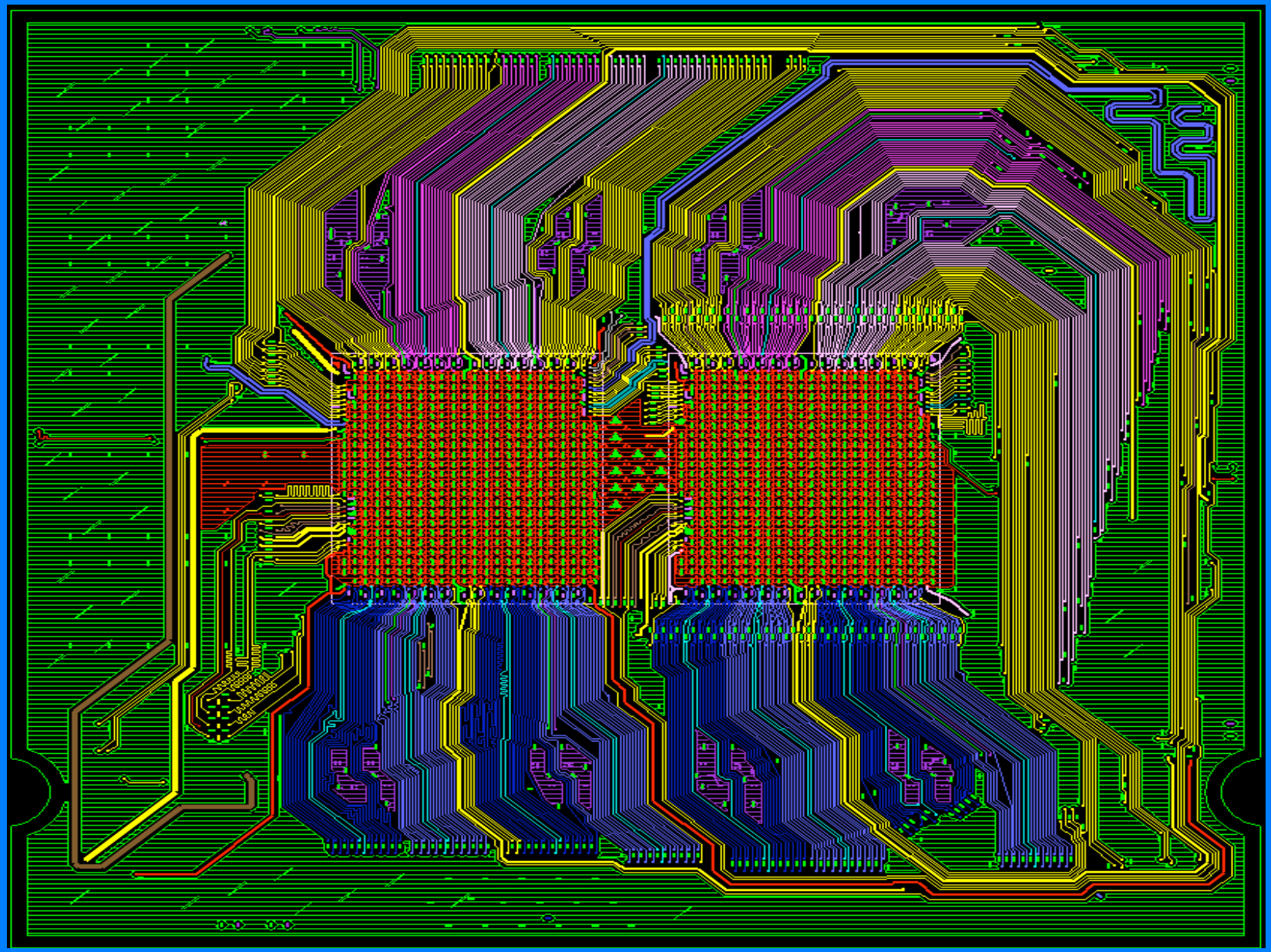
1) Intel® Pentium® 4 Processor with HT Technology Extreme Edition 3.73GHz (2 MB L2 Cache, 1066 MHz FSB) and Intel® 925XE Express Chipset
2) Intel® Pentium® Processor Extreme Edition 840 (2x1 MB L2 Cache, 3.20 GHz, 800 MHz FSB, HT Technology) and Intel® 955X Express Chipset

Challenges in migrating to multi-core

- Rapid movement from single core design to multi-core design presented many complexities
  - Already existing platform hardware
  - Factory already populated with manufacturing hardware
  - Test database developed for single core
  - Tight package dimensions
  - Little power headroom left

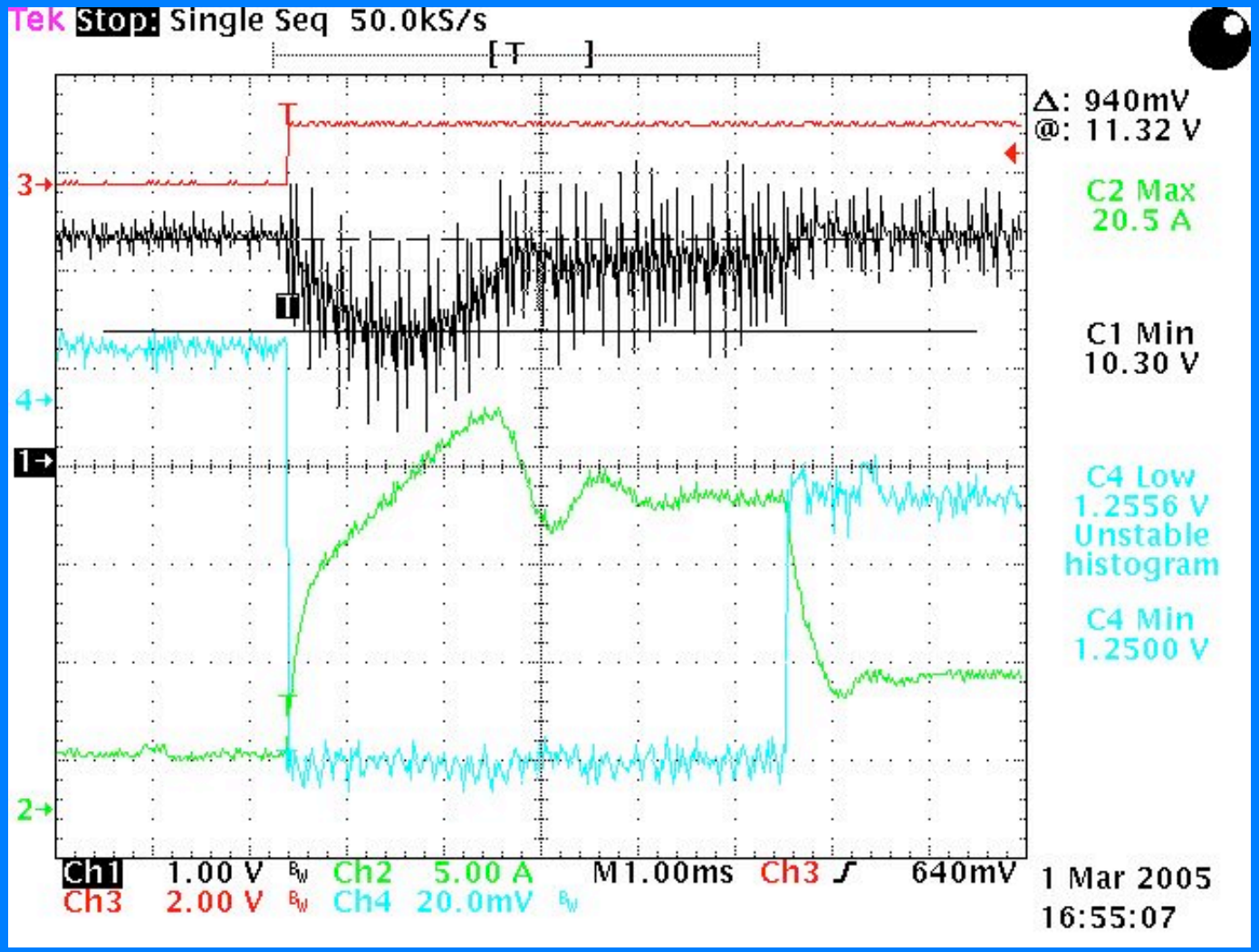
Package issue

- Package design a huge challenge
  - More layers required (Just address/data alone is > 100 more signals)
  - Same package cavity and pinout – couldn't grow.
  - New IHS (Integrated Heat Sink) required for thicker package
  - Power cap placement can't be centered over both cores
  - Existing signals on 4 sides of core causes power bus routing voids.
- No logic outside core. Any needed logic must be in core. Lots of 'special signal' headaches like thermal diode, ODT (On-Die Termination).



Power constraints

- Existing platform dictated 1 power plane for both cores
  - Penalized for 2X leakage, required architecting a speed-step protocol
- 2 cores powering up & fully active cause large di/dt events
  - Required Voltage Regulator mods to grow headroom to 125A plus silver box restrictions
- Required BIOS change to boot to low voltage/frequency on performance parts.
  - BIOS initiates speedstep event to all threads after completion

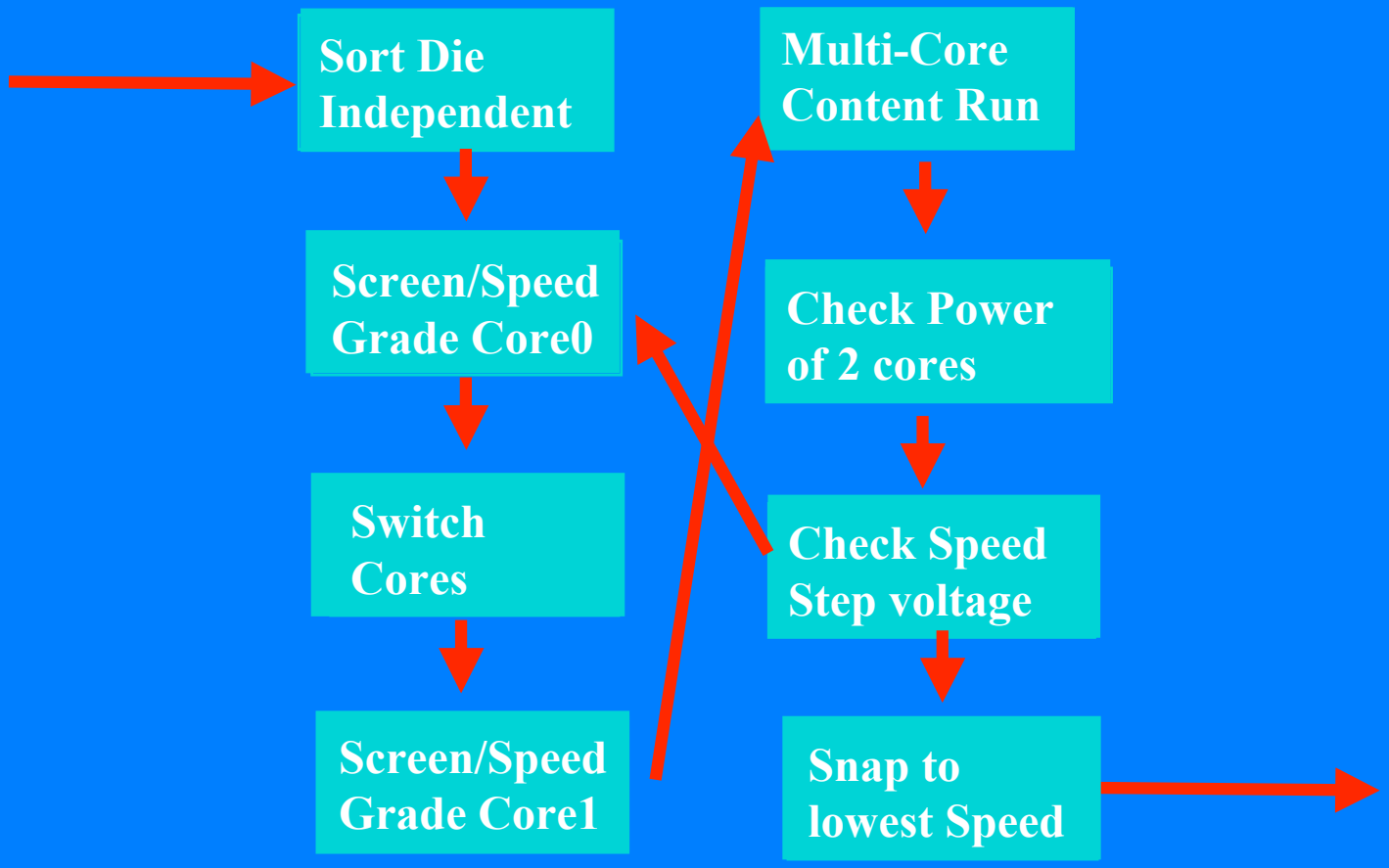


2-core boot to full speed, weak power supply

Test issues

- Thousands of hours invested in single core coverage database
  - Copied core design a plus
  - Needed to add ‘core swap & kill’ hardware to reuse database
- Existing single core test can’t expose problems on core->core interaction
  - Voltage transients, thermal gradient
  - Some explicit dual core content required

Test flow example



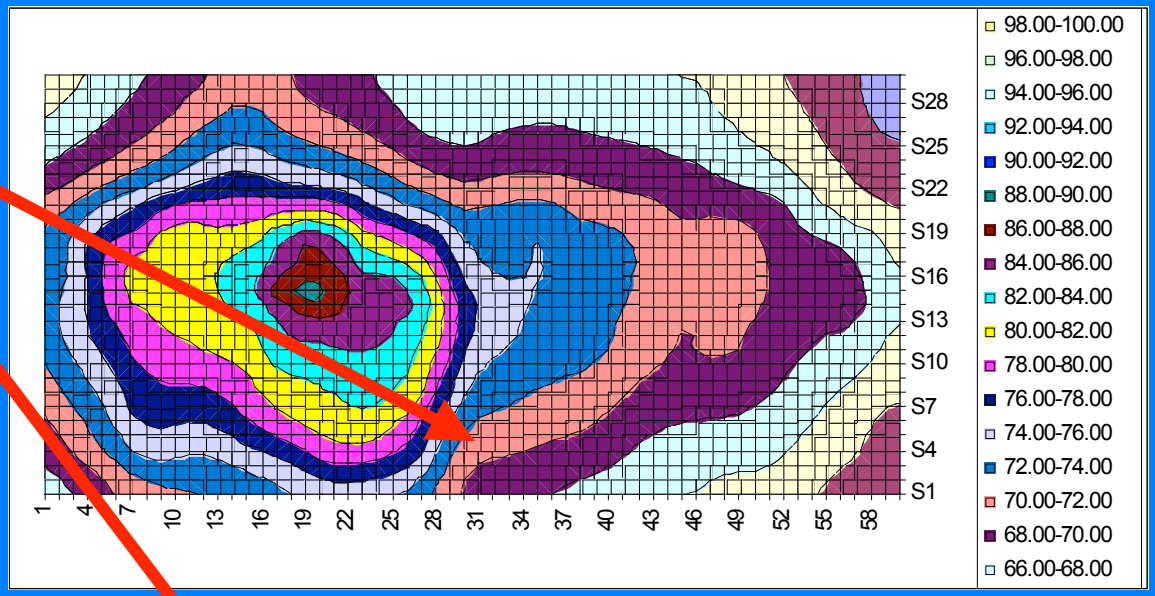
Thermal issue

- Platforms support only 1 ADC for thermal monitoring
  - 2 cores can create many different thermal profiles
  - Diode temp to junction hot spot delta can vary depending on workload & core utilized
- Required thermal protection to be independent on both cores

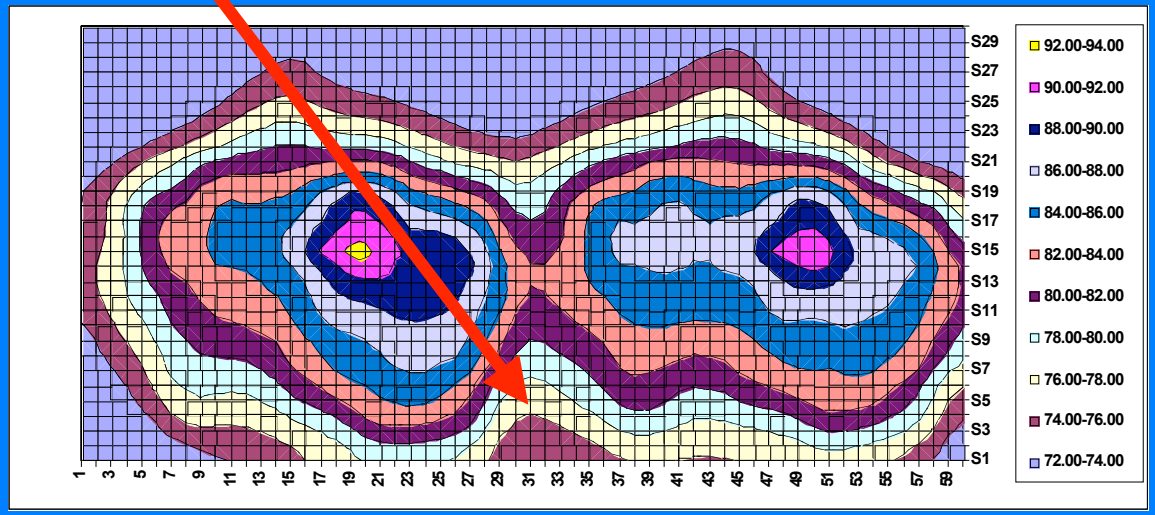
Thermal gradients

Thermal Diode

Core0 only



Dual Core



Limitations of shared bus

- 2 loads on bus = less bus speed.
  - Plus 1M cache = more bus traffic. Double whammy.
- Difficult package design
  - ~2x traces to same number pins
- Thermal & electrical properties degrade.
  - Slow down penalizes both cores.
- Segregated die.
  - Test overhead. Slowest die constrains final product.

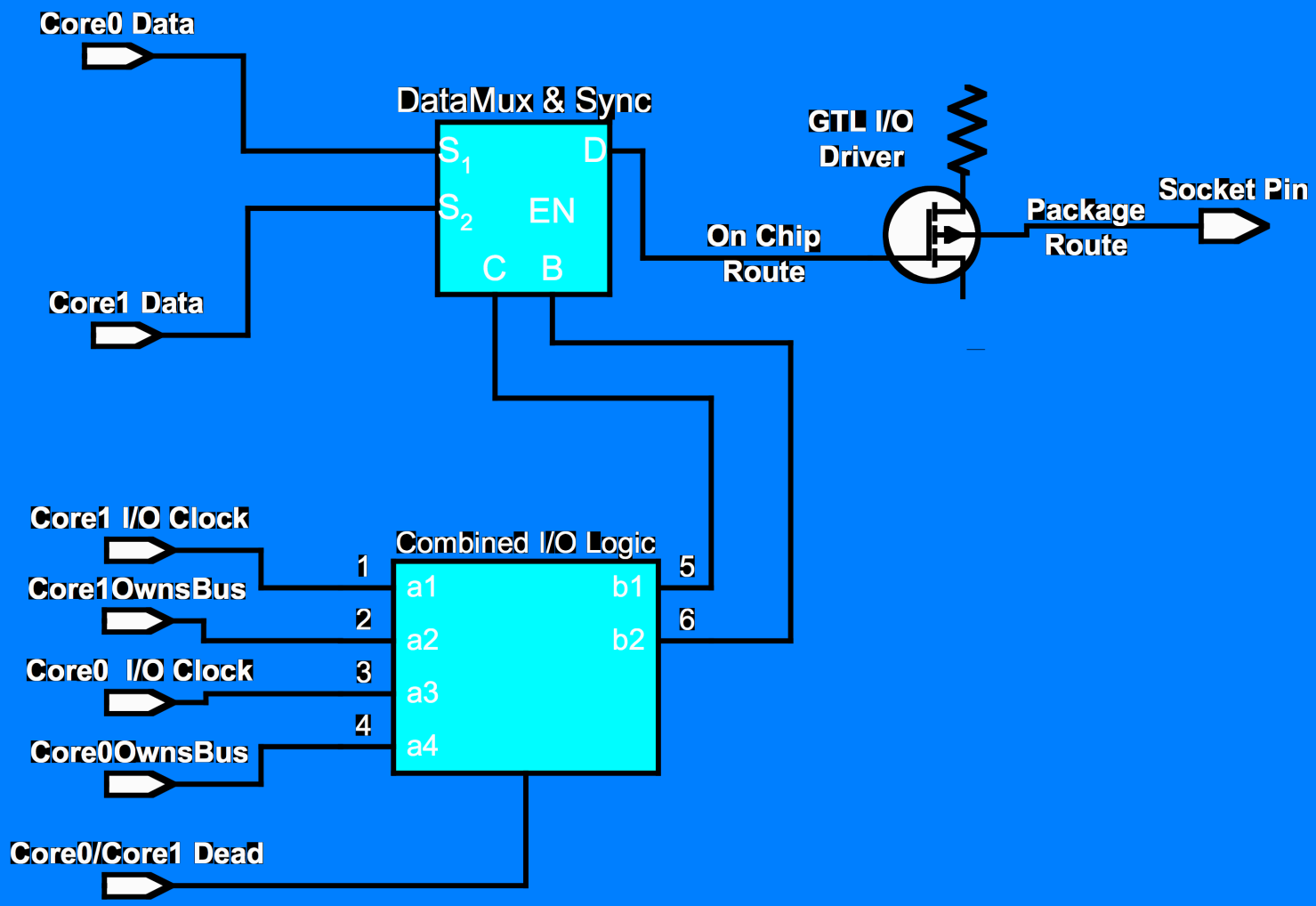
Overview of Paxville-MP processor

- Dual-Core/Multi-Threaded Xeon Processor on 90nm process
  - 2-2M caches, 667Mhz min FSB, up to 4 threads.
  - Platform still 4-P compatible for up to 16 threads per platform
- Dual bus platform – 2 CPU agents per bus
  - Only 1 load presented to system by CPU
- Enhanced auto-halt and 2-state speed step power management
  - Independent events supported per core.

Advantages of new Paxville design

- Single CPU load on bus. Allows faster bus, less electrical load.
  - 8 agents (16 threads) on top end platform
- Larger cache = less FSB bottlenecks
- Better package design
  - Fewer traces allows better power delivery
- Integrated die (monolithic)
- Consolidated bus logic allows test enhancements

Paxville consolidate bus



Challenges with Paxville design

- Degraded I/O timing with shared bus
  - Requires extra logic & routing but must be compatible to existing bus timing.
  - Requires circuit tricks for quad pumped bus.
- Enhancements to validation tools
  - 8xx series treated as 2 independent CPUs. Paxville is integrated – 1 die.
- Additional complexity in test infrastructure.
  - New test modes & consolidated bus logic.

Going forward with multi-core

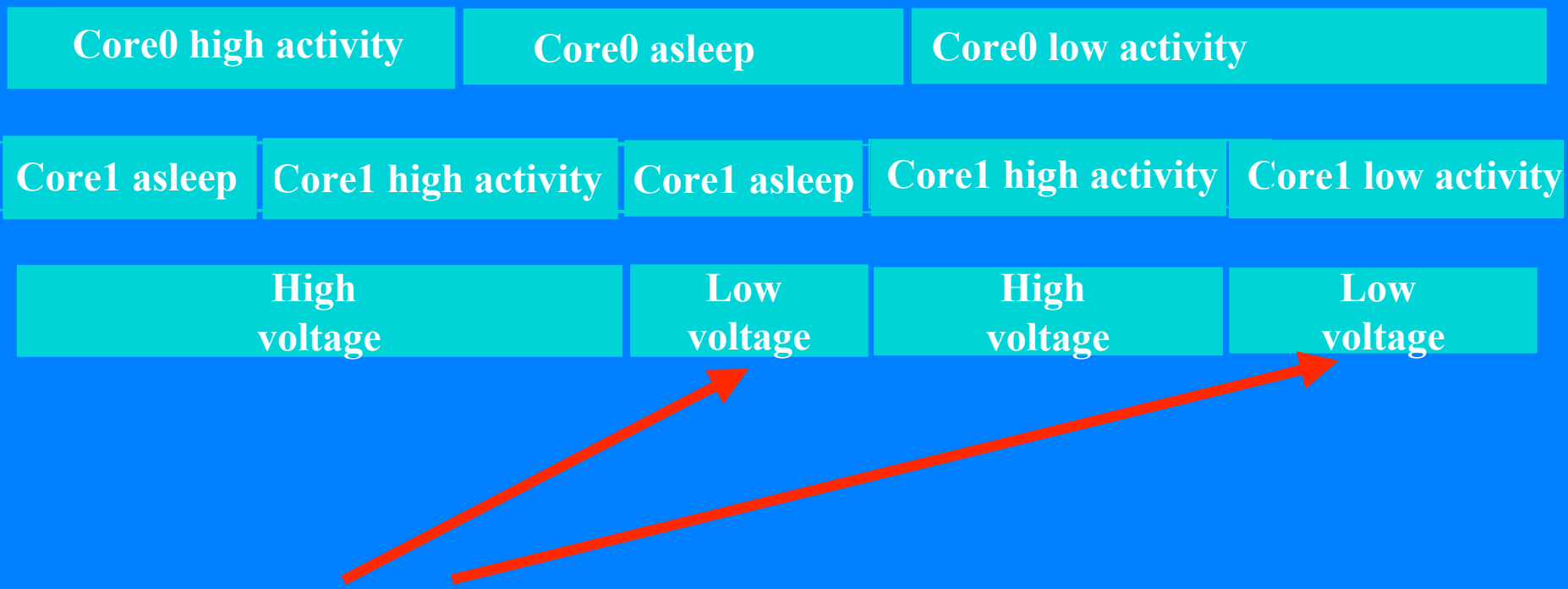
- Solving bus bottlenecks.
- Integrate next level cache for less bus traffic.
  - Downside is higher latency on cache misses.
  - Upside is lower pin count & can stay with a flexible bus architecture
  - Cache thrashing by multiple cores an issue if size isn't large enough – swamps bus again.
- 'Point-to-point' busses & memory controllers
  - Upside is no bus traffic collisions
  - Downsides are being locked into memory protocol and a huge pin count increase.

Going forward with multi-core

- Solving power issues..
- Need better power state management
  - Single voltage plane is an issue – can't drop leakage on inactive cores
  - Need more intelligence in controller
- Segment products with power in mind
  - Typically done more now on speed/feature set.
  - Can microprocessor be 'tuned' for a power segment.

SpeedStep protocol

Core Activity over time



Limited opportunities to reduce power, much harder with even more cores

Going forward with multi-core

- Core counts will continue to increase.
  - Higher threaded applications give opportunity to have better power / performance.
  - Power is wasted when a core that isn't working on a thread is alive, but performance is wasted if OS has to continually swap out threads.
- Expect that logic to 'glue' cores together will become as critical as the core
  - Need lots of sophistication to take full advantage of a high core count
  - Need busses capable of handling the high traffic to memory